15-424/15-624 Recitation 12
Game proofs

1. **Dual operators**

   The dual operators of differential game logic are particularly sneaky. They'll let you think they're easy and trivial, them **BAM**, they'll hit you with a particularly gnarly stick. That's a true story if there ever was one.

   What went wrong with the following proof?

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{*}{\vdash 0=0}}{\vdash [x:=0]x=0}[:=] \quad \dfrac{\vdash 1=0}{\vdash [x:=1]x=0}[:=]}{\vdash [x:=0\cup x:=1]x=0}[\cup]}{\vdash [(x:=0\cup x:=1)^*]x=0}\text{ind}}{\vdash \langle (x:=0\cup x:=1)^\times\rangle x=0}\langle^d\rangle$$

   The end property says that Angel wants to get $x=0$, and she can achieve that in each iteration of the loop, but Demon can decide to repeat. We saw that he isn't allowed to repeat indefinitely, so Angel should win.

   What happened is that when converted the Demon loop into an Angel loop, we forgot that there were a lot more duals hanging around. For this reason, we highly encourage you to rewrite all programs using *only* dual operators, and removing all Demon-specific notation like $\alpha\cap\beta$ and $\alpha^\times$.

   In this case, we'd get:

$$(x:=0\cup x:=1)^\times \equiv ((((x:=0)^d\cup(x:=1)^d)^d)^*)^d$$

   Since assignment is deterministic, that reduces to:

$$(((x:=0\cup x:=1)^d)^*)^d$$

   So we had actually forgotten a few dual operators. Silly TAs. Because we fully understand dual operators, the proof now has *no chance* of going wrong! *FACT!*

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{*}{\vdash 0=0,\langle x:=1\rangle x=0}}{\vdash \langle x:=0\rangle x=0,\langle x:=1\rangle x=0}[:=]}{\vdash \langle x:=0\cup x:=1\rangle\, x=0}\langle\cup\rangle]}{\vdash \left[((x:=0\cup x:=1)^d\right] x=0}[^d]}{\vdash \left[\left((x:=0\cup x:=1)^d\right)^*\right] x=0}\text{ind}}{\vdash \left\langle\left(\left((x:=0\cup x:=1)^d\right)^*\right)^d\right\rangle x=0}\langle^d\rangle$$

2. **Convergence**

With loops inside box modalities, we know we can use invariants. Using invariants for loops inside diamond-modalities is overkill since the property only needs to hold at the end of one of the iterations.

This is relevant because Angel needs to be able to do something easily about loops!

So we have a convergence rule, which states that as long as we keep making progress towards a point at which our property is satisfied, we are good! To this end, we'll be playing a game we like to call "get to 0". It involves, unsurprisingly, getting to 0!

$$\text{con'}\,\frac{\Gamma \vdash \exists v.\varphi(v) \qquad \vdash \forall v \geq 0.\varphi(v) \to \langle\alpha\rangle\varphi(v-1) \qquad \exists v \leq 0.\varphi(v) \vdash \phi}{\Gamma \vdash \langle\alpha^*\rangle\phi, \Delta}$$

The idea here is that we'll try to prove a convergence property φ. If ever this property holds for a negative number, it has to imply our desired property ϕ. So all we have to prove is that it makes "progress" towards, and past 0!

Let's examine each branch and see what it says (I'm giddy with excitement already)!

(a) In the conditions under which we start, Γ, we can prove that there's some number for which a convergence property φ holds. If the number is negative, awesome. If it's not, then we move on to the next branch!

(b) If φ holds for any positive number v, then it will hold for $v-1$ after one iteration of the loop. In other words, by continuously iterating the loop, which Angel can easily choose to do, she can get φ to hold for a negative value!

(c) The concluding branch! If the property holds for some negative value (and we ensured it does with the two previous branches), then the desired property ϕ can be proved.

The problem now, like for invariants, is coming up with the convergence property! Let's start with the simple example from lecture.

$$x > 0 \vdash \langle(x := x-1)^*\rangle x < 1$$

Notice how we want to get x to be smaller than 1 specifically. It helps us to think backwards form the third branch. What kind of property for negative v guarantees that $x < 1$? Well, if $x < v+1$, then in the worst case scenario, v would be 0, and we'd have our property $v < 1$. If n was any smaller, we'd just be making it easier on ourselves! So we use that as a candidate property.

$$\text{con'}\,\frac{\mathbb{R}\,\dfrac{*}{\vdash \exists v.x < v+1} \qquad \forall_r\,\dfrac{\to_r\,\dfrac{\langle:=\rangle\,\dfrac{\mathbb{R}\,\dfrac{*}{v \geq 0, x < v+1 \vdash x-1 < v}}{v \geq 0, x < v+1 \vdash \langle x\text{--}\rangle x < v}}{v \geq 0 \vdash x < v+1 \to \langle x\text{--}\rangle x < v}}{\vdash \forall v \geq 0.x < v+1 \to \langle x\text{--}\rangle x < v} \qquad \text{QE}\,\dfrac{*}{\exists v \leq 0.x < v+1 \vdash \phi}}{\vdash \langle(x\text{--})^*\rangle x < 1}$$

That worked out! Aren't we glad we have QE!

3. **A more complex con' example**

So now that we've got a full understanding of the simpler proof under our belts, let's try using the con' rule to prove a more complex game.

$$\vdash \langle x := 0; (x' = -1)^d; (x := x + 1)^* \rangle x > 0$$

So Angel lets Demon make x as small as he wants to. She maybe even goes make a cup of tea while this is happening - who knows! Anyway, when she comes back, x is really super small. The good news is that Angel is in heaven, which is really one big permanent holiday, so she gets to keep adding 1 to x, however many times she wants. And she wants to make sure that x becomes positive again, not negative like before.

We can use $x > -v$ as our convergence property, and start with decomposing our problem.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{t \geq 0 \vdash \{x := -t\}\langle (x := x + 1)^* \rangle x > 0}\textit{ to be continued...}}{t \geq 0 \vdash \{x := 0\}[x := x - t]\langle (x := x + 1)^* \rangle x > 0}[:=]}{\vdash \{x := 0\}\forall t \geq 0.[x := x - t]\langle (x := x + 1)^* \rangle x > 0}\forall_r, \rightarrow_r}{\vdash \{x := 0\}[x' = -1]\langle (x := x + 1)^* \rangle x > 0}\text{ODE}}{\vdash \{x := 0\}\langle (x' = -1)^d \rangle \langle (x := x + 1)^* \rangle x > 0}\langle^d\rangle}{\vdash \langle x := 0 \rangle \langle (x' = -1)^d \rangle \langle (x := x + 1)^* \rangle x > 0}\langle:=\rangle}{\vdash \langle x := 0 \rangle \langle (x' = -1)^d; (x := x + 1)^* \rangle x > 0}\langle;\rangle}{\vdash \langle x := 0; (x' = -1)^d; (x := x + 1)^* \rangle x > 0}\langle;\rangle$$

There are a couple of cool things happening! First of all, notice how we applied some rules to inner formulas instead of only to the outer-most formula! We are now allowed to do that because we had equivalences instead of proof rules (which only go one way). Second, we get to have fun carrying a delayed assignment around! *YAY!*

Where do we carry the delayed assignment to though? Every branch? The key issue here is that the assignment only applies to the original variables, i.e. the ones that Γ mentioned. It gets passed around as context - if the context isn't there because of soundness, then neither should the delayed assignment!

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{*}{t \geq 0 \vdash -t > -t - 1}\mathbb{R}}{t \geq 0 \vdash \exists v. -t > -v}\exists_r v \mapsto \text{-}t\text{-}1}{t \geq 0 \vdash \{x := -t\}\exists v.x > -v}\{:=\}}{t \geq 0 \vdash \{x := -t\}\langle (x := x + 1)^* \rangle x > 0}\text{con'} \qquad \text{The big branch!} \qquad \text{QE}\dfrac{*}{\exists v \leq 0.x > -v \vdash x > 0}$$

And finally, the last branch is very similar to the one from the simple example.

$$\forall_r \frac{\to_r \frac{\langle:=\rangle \frac{\mathbb{R} \frac{*}{v \geq 0, x > -v \vdash x + 1 < -v + 1}}{v \geq 0, x > -v \vdash \langle x{+}{+}\rangle x > -v + 1}}{v \geq 0 \vdash x > -v \to \langle x{+}{+}\rangle x > -v + 1}}{\vdash \forall v \geq 0. x > -v \to \langle x{+}{+}\rangle x > -v + 1}$$

And thus is the proof concluded, and the Angel hath won this most epic Battle of the Tea & Holidays.